



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

INTERFACE WEB PARA SISTEMA DE GESTÃO DE REDES SEGUINDO PROJETO CENTRADO EM USUÁRIO

Rafael M. P. Chianca

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia de Computação

Orientador
Prof. André Costa Drummond

Brasília
2020

Agradecimentos

Obrigado ao professor André Costa Drummond por sua disponibilidade, paciência e atitude positiva. Essas foram características que fizeram dele um dos melhores professores que tive em meus cinco anos de graduação e definitivamente foram essenciais no decorrer do desenvolvimento desta monografia. Obrigado ao Rafael Alves por sempre estar disposto à responder perguntas relacionadas aos diversos elementos da rede, bem como a testar a usabilidade do sistema e ao Renato Araújo por gerir o desenvolvimento do sistema. Um grande obrigado aos meus colegas e amigos Lucas Campos, João Marcelo Nunes e Andre Garrido que também estiveram presentes neste projeto.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Com o crescimento da infraestrutura da rede gerida pela Associação GigaCandanga, surgiu uma demanda por um sistema que fosse fácil de ser usado pelos gestores e, ao mesmo tempo, robusto suficiente para conter todas as informações necessárias a respeito das redes em questão. A instituição fazia a gerência das redes sob sua supervisão por meio da utilização extensiva de planilhas. Manter a gestão de uma quantidade tão abundante de elementos que compõem uma rede desta escala, por meio de planilhas provou-se complexo e, em alguns casos, inseguro, por diversos fatores como: erro humano no preenchimento de informações, redundância de dados, desorganização ou falta de padronização de campos referentes aos elementos. A partir destes problemas apresentados ao longo da utilização do antigo método de gestão, surgiu a ideia de desenvolver um sistema dedicado à gerência de redes de fibras óticas e seus elementos. Visando a melhor usabilidade do sistema, bem como uma boa experiência de usuário, o desenvolvimento deste sistema contou com a utilização de diversas práticas adotadas no mercado de desenvolvimento de software em relação ao desenvolvimento de interfaces centrado no usuário.

Palavras-chave: Front-end, react, interface de usuário, experiência de usuário, design centrado no usuário

Abstract

With the constant expansion of the network managed by the GigaCandanga institution, the need for a system capable of storing and manipulating great amounts of data, as well as being very user friendly grew. Previously to the development of this system, said institution managed its networks with the use of a great amount of spreadsheets. Maintaining the great amount of data on spreadsheets proved to be quite a complex and unreliable job, because of factors such as: human error filling out the spreadsheet's data, lack of organization, lack of patterns regarding naming and so on. Because of these many problems the institution has encountered over its many years of network management, the idea of developing a software dedicated to the management of networks and all its elements. To ensure this software is user friendly and provides a good user experience, the development of this system was based upon multiple user-centered design practices such as user testing and constant user feedback, as well as many other practices discussed in this document.

Keywords: Font-end, react, user interface, user experience, user-centered design

Sumário

1	Introdução	1
1.1	Objetivos	2
1.2	Soluções Obtidas	3
1.3	Organização do Documento	3
2	Conceitos Básicos	5
2.1	Elementos da Rede	5
2.2	Design Centrado no Usuário	6
2.2.1	Experiência de usuário	7
2.3	React	8
2.3.1	Por Que Usar React?	8
2.3.2	Componentes	9
2.3.3	Estados	10
2.3.4	Props	10
2.4	Leaflet	11
2.4.1	Eventos	11
2.5	Metodologia Ágil	12
2.5.1	Scrum	12
2.6	Resumo Conclusivo	14
3	Metodologia	15
3.1	Definição de Objetivos	15
3.1.1	Funcionalidades	16
3.1.2	Requisitos do Usuário	16
3.2	Prototipação	17
3.3	Execução das Sprints	19
3.3.1	Demonstração das funcionalidades	19
3.4	Resumo Conclusivo	19

4	Implementação	21
4.1	<i>Pop-ups</i>	21
4.1.1	Criação	21
4.1.2	Edição	23
4.1.3	Outras Funcionalidades	23
4.2	Tabelas	24
4.2.1	Filtragem por Geolocalização	25
4.2.2	Adição de Elementos	25
4.2.3	Edição de Elementos	26
4.3	Mapa	26
4.3.1	Visualização de Elementos	27
4.3.2	Criação de Elementos	29
4.3.3	Edição de Elementos	30
4.3.4	Exclusão de Elementos	30
4.4	Resumo Conclusivo	31
5	Resposta dos Usuários	32
5.1	Métricas de Experiência do Usuário	32
5.1.1	Taxa de Conversão	32
5.1.2	<i>Net Promoter Score (NPS)</i>	33
5.1.3	Taxa de Sucesso de Tarefas	33
5.1.4	Escala de Usabilidade do Sistema	33
5.1.5	Escala de Dificuldade de Tarefa	34
5.2	Avaliação do Sistema	35
5.2.1	Navegação	35
5.2.2	Importação de Arquivos	36
5.2.3	Interface de Tabelas	36
5.2.4	Interface de Mapa	37
5.2.5	Avaliação da Escala de Usabilidade	38
5.2.6	Erros Encontrados	39
5.3	Resumo Conclusivo	40
6	Conclusão	41
	Referências	43

Lista de Figuras

1.1 Visão geral do sistema desenvolvido, apresentando a separação entre as frentes de desenvolvimento.	2
2.1 Exemplo fictício da rede da GigaCandanga.	6
3.1 Protótipo da página de tabela de segmentos, com opção de criar, editar, e excluir segmentos.	18
3.2 Protótipo do <i>pop-up</i> de mapa utilizado para escolher localização dos elementos.	18
4.1 <i>Pop-up</i> de criação de <i>jumper</i> . Este formulário contém seletores para fazer a associação com outros elementos da rede.	22
4.2 <i>Pop-up</i> de criação de caixa de emenda. Este formulário contém campos numéricos e de texto.	22
4.3 <i>Pop-up</i> de seleção de trechos de cabo à serem associados. Linhas azuis representam trechos de cabo selecionados pelo usuário.	24
4.4 Ícone associado à filtragem por geolocalização.	25
4.5 Visão geral da interface de mapa. Os ícones ao lado esquerdo da imagem são funcionalidades de manipulação do mapa e de elementos nele mostrados. No canto superior direito está presente o menu seletor de <i>layout</i> de mapa. No centro do mapa- em cor preta- estão representados trechos de cabo, caixas de emenda e postes.	27
4.6 Layout do mapa utilizando a API do Google Maps Roads.	28
4.7 Layout do mapa utilizando a API do Google Maps Satellite.	29

Lista de Tabelas

- 5.1 Notas referentes às afirmativas de escala de usabilidade do sistema. A coluna esquerda apresenta o número da afirmativa em questão. A coluna do meio apresenta os valores atribuídos pelo usuário. A coluna direita apresenta os valores obtidos após a aplicação da lógica mencionada na Seção 5.1.4. . . . 39

Lista de Abreviaturas e Siglas

CRUD Create, Read, Update and Delete.

CSS Cascading Style Sheets.

DCU Design Centrado no Usuário.

DGO Distribuidor Geral Óptico.

HTML Hypertext Markup Language.

IU Interface de Usuário.

JS JavaScript.

MCT Ministério da Ciência e Tecnologia.

NPS Net Promoter Score.

RNP Rede Nacional de Ensino e Pesquisa.

UCD User-Centered Design.

UX User Experience.

Capítulo 1

Introdução

Este trabalho foi baseado na descrição detalhada de um sistema desenvolvido à fim de possibilitar a gerência de uma rede de fibras óticas e outros elementos necessários para o funcionamento desta. Este sistema surgiu a partir da necessidade da Associação GigaCandanga [1] de adotar o uso de um sistema capaz de visualizar, manter e gerir, de maneira simples e eficaz, a infraestrutura da rede de fibras óticas que a instituição em questão gerencia. A Associação GigaCandanga é responsável pela manutenção, operação e gestão da rede metropolitana do Distrito Federal, Rede GigaCandanga, criada no âmbito da iniciativa intitulada Redecomep, do Ministério da Ciência e Tecnologia (MCT), coordenada pela Rede Nacional de Ensino e Pesquisa (RNP), que tem como objetivo implementar redes de alta velocidade nas regiões metropolitanas do país servidas pelos Pontos de Presença da RNP.

O trabalho apresentado é um esforço em conjunto com outros três alunos do curso de engenharia de computação, com o propósito final da implementação de um sistema para o cadastro, gestão e manutenção de uma rede de fibras óticas da associação GigaCandanga. Este trabalho visa descrever o processo de desenvolvimento da interface do sistema de redes, bem como as técnicas utilizadas para tal feito. Para uma visão completa da solução, é sugerida a leitura de todos os trabalhos: Chaves[2](2020), Damaceno[3](2020) e Jorge[4](2020).

No ponto de vista do desenvolvimento, foram criadas duas equipes, uma com o propósito de desenvolver a interface do usuário e outra de desenvolver a aplicação de manipulação dos dados do sistema de gestão da rede, denominadas de *front-end* e *back-end* respectivamente.

Uma esquematização geral do sistema pode ser observada na Figura 1.1. Nela se observa uma visão ampla do funcionamento da aplicação, bem como as conexões dos módulos que resultam no funcionamento total do sistema.

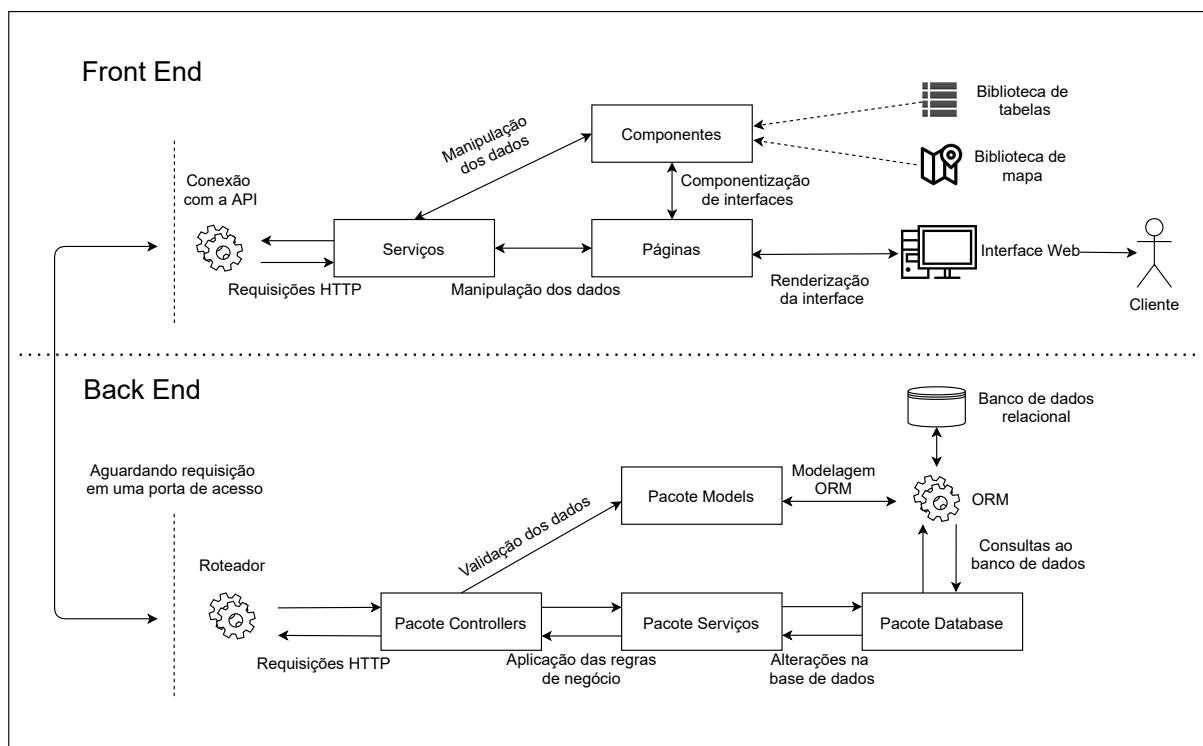


Figura 1.1: Visão geral do sistema desenvolvido, apresentando a separação entre as frentes de desenvolvimento.

1.1 Objetivos

Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de uma interface para um sistema de gestão de redes de fibras óticas por meio da adoção de práticas de design centrado no usuário.

Objetivos Específicos

A partir da definição do objetivo geral, é possível definir objetivos específicos:

- Criar elementos contendo latitude e longitude como atributos.
- Representar elementos em um mapa.
- Representar segmentos de reta no mapa de maneira fidedigna a fim de representar trechos de cabo.
- Manipulação (criação, edição e exclusão) de elementos através do mapa.

- Implementar páginas dedicadas à cada tipo de elemento, possibilitando a execução do CRUD¹ dos elementos.
- Avaliar, a partir do uso de métricas, a experiência do usuário e a usabilidade do sistema.

1.2 Soluções Obtidas

O sistema final é capaz de fazer o gerenciamento de elementos físicos que compõem a rede, além de uma fácil maneira de visualizar a rede como um todo por meio de uma interface amigável de mapa.

Por meio da utilização de páginas dedicadas à cada elemento, o usuário consegue visualizar, em forma de tabelas, os elementos cadastrados, bem como filtrá-los por localização (quando o elemento tiver esse tipo de atributo) ou por qualquer outro campo que estes elementos possuam. Estas páginas dedicadas facilitam também a exclusão em massa de elementos, bem como a edição de campos específicos destes, por meio de formulários.

A interface de mapa implementada permite a visualização, edição e exclusão de diversos tipos de elementos, além da opção de criar novos elementos a partir de sua posição no mapa, não havendo necessidade de preencher coordenadas geográficas manualmente, caso o usuário deseje esta funcionalidade.

Outra funcionalidade implementada no sistema é a importação de arquivo de extensão kmz (*keyhole markup zipped*) ou kml (*keyhole markup language*), que são arquivos compostos por diversas localizações geográficas para diferentes tipos de elementos da rede gerenciada. Este tipo de arquivo é associado ao software Google Earth [5], o que faz a implementação da funcionalidade de importação ainda mais importante, visto que a gestão da rede se baseia fortemente na utilização deste software. Além disto, a importação deste tipo de arquivo facilita a transição entre a antiga gestão da rede para a gestão feita com o sistema descrito por este documento, bem como facilita qualquer futura importação de dados caso já existam arquivos .kmz ou .kml prontos para uso.

1.3 Organização do Documento

O Capítulo 2 deste documento descreve com maior detalhamento os conceitos necessários para o entendimento do processo de desenvolvimento do sistema, tal como a terminologia utilizada e as tecnologias escolhidas (e o porquê destas escolhas). Em sequência, no Capítulo 3, é apresentada a metodologia adotada para o desenvolvimento do sistema,

¹Termo original em inglês *Create, Read, Update and Delete*, significando, respectivamente: criar, ler, editar e deletar.

bem como uma descrição sobre o processo de organização do desenvolvimento em si. O Capítulo 4 é referente ao processo de implementação do código que compõe o sistema, explicando o funcionamento de cada página que compõem a interface, a maneira como foram implementadas e suas peculiaridades. No Capítulo 5 é explicado, de maneira detalhada, o processo de teste da interface com o usuário final e os resultados obtidos por meio destes. Por fim, no Capítulo 6 é feita a análise dos objetivos definidos na Seção 1.1, comparando estes com os resultados finais do desenvolvimento do sistema, além da conclusão do trabalho realizado.

Capítulo 2

Conceitos Básicos

Este capítulo descreve os conceitos necessários para entender o objeto de estudo deste documento, visando um fácil entendimento para leitores que não estão familiarizados com as tecnologias utilizadas, ou até mesmo com programação de uma maneira geral. Os conceitos que serão apresentados incluem conceitos sobre elementos de rede, conceitos gerais de programação, conceitos de programação específicos ao contexto do desenvolvimento do sistema de gestão, descrições da tecnologia utilizada e métodos de organização de desenvolvimento.

2.1 Elementos da Rede

Para o pleno entendimento deste trabalho, é necessário antes entender as necessidades que a instituição mencionada possui para a gerência da rede de fibras óticas. Como explicado no Capítulo 1, a instituição GigaCandanga é responsável pela gestão de uma rede de fibras óticas. Esta rede é composta por diversos elementos, sendo a maioria deles físicos, com a exceção de alguns elementos que podem ser caracterizados como elementos à nível lógico.

A rede em sua maioria é composta por cabos de fibra ótica, que podem ser caracterizados como filamentos flexíveis (geralmente de vidro) que transmitem dados em forma de luz. Estes cabos possuem um comprimento finito e, por isso, existe a necessidade de emendá-los por meio de outro elemento físico chamado de caixa de emenda, podendo ser descrito como uma caixa onde ocorre a conexão dos cabos, por meio de um processo de fusão que é feito fibra a fibra. As caixas de emenda podem ser categorizadas como aéreas ou subterrâneas, sendo a caixa aérea atrelada a um poste, que é outro elemento físico que o sistema deve manter registro. Nos postes também pode existir outro elemento denominado reserva técnica- um excesso de cabos reservados para uma eventual necessidade, como um reparo ou uma expansão da rede. O objetivo final da conexão entre estes elementos é prover acesso ao que são denominadas Instituições Parceiras. Dentro

das instituições parceiras são encontrados locais onde os equipamentos da GigaCandanga estão armazenados, chamados de *sites*. Os *sites* por sua vez, podem ser denominados como concentrador- um site que serve como ponto de distribuição para os demais locais da rede- ou de acesso, servindo apenas para a instituição parceira utilizar como ponto de conexão com seus próprios dispositivos. Além disto, os *sites* possuem painéis nos quais as fibras de um cabo são separadas para poderem dar acesso aos equipamentos da instituição, que são chamados de distribuidores gerais ópticos (DGO), e *switches*, que é o equipamento utilizado para a conexão e filtragem de informações entre duas ou mais estações de trabalho em uma rede de computadores. Por último, temos o elemento de nível lógico, chamado de segmento, caracterizado por ser uma junção de todos elementos que conectam uma instituição à outra.

A Figura 2.1 é um exemplo parcial da rede, onde os segmentos são representados pelas arestas coloridas, as caixas de emenda pelas formas retangulares e as instituições parceiras por cilindros contendo o nome da instituição à qual se refere.

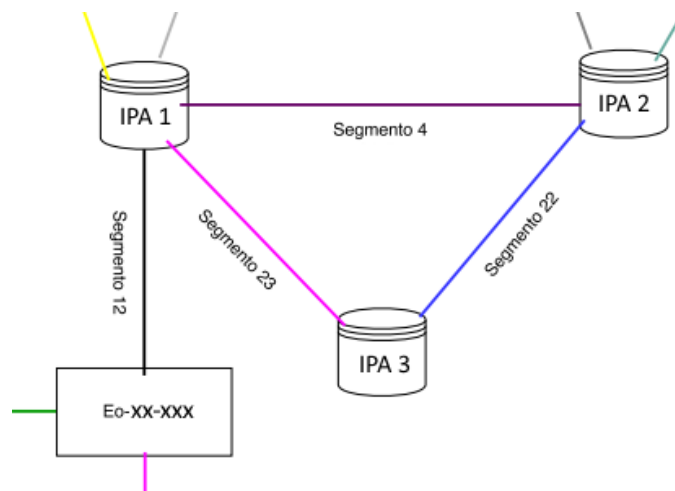


Figura 2.1: Exemplo fictício da rede da GigaCandanga.

2.2 Design Centrado no Usuário

O termo *Design Centrado em Usuário*, do inglês *User-Centered Design (UCD)*, se refere ao processo de criar o *design* de uma aplicação ou sistema (pode se aplicar a mais campos do que somente ao da programação) de maneira que o foco deste é a fácil usabilidade por seus usuários alvo. O objetivo final do DCU, especialmente no contexto de desenvolvimento de software, é garantir não só a melhor usabilidade possível da interface desenvolvida, mas também garantir que este processo seja feito de uma maneira mais eficaz, evitando o retrabalho e a necessidade de gastar tempo com problemas que podem surgir durante

tal desenvolvimento. Um exemplo frequentemente encontrado no processo de desenvolvimento de qualquer sistema é: a necessidade de novas reuniões para definição de requisitos e funcionalidades que não possuem real valor para o sistema em questão e funcionalidades que deveriam ter sido implementadas desde o início do processo de desenvolvimento e não foram.

Para o *Design Centrado em Usuário* ser executado da melhor maneira possível, é necessário implementar determinadas práticas de desenvolvimento, além de práticas de planejamento de projeto. Como listado em Lowdermilk[6](2013), existem diversas etapas de planejamento ao se trabalhar em equipe, que podem ser traduzidas (tradução livre)¹ em:

- Detalhar o projeto;
- Listar requisitos do usuário;
- Listar funcionalidades;
- Criar diagramas do fluxo de trabalho e/ou do banco de dados;
- Criar imagens de protótipos.

Seguir estas etapas significa garantir para o projeto uma base de organização no processo de desenvolvimento da interface de usuário, bem como minimizar o tempo gasto em um futuro retrabalho de *design* e programação para consertar ou adicionar funcionalidades que deveriam ter sido executadas desde o princípio do processo. Conceitualmente o DCU é simples, porém, ao longo do desenvolvimento deste documento será melhor exposta a importância da implementação das práticas descritas para o melhor desenvolvimento do sistema de gerenciamento de redes.

2.2.1 Experiência de usuário

Experiência de usuário (EU), ou *User Experience (UX)* do inglês, é um termo utilizado para descrever o aspecto interativo humano-computador de uma dada interface de usuário. Este conceito é muito atrelado ao Design Centrado no Usuário, uma vez que o processo de design de uma interface baseado em seu usuário, tem como objetivo final uma interface robusta e de fácil usabilidade, garantindo uma ótima experiência ao usuário.

O termo experiência de usuário é constantemente confundido com o conceito de usabilidade, uma vez que ambos conceitos realmente estão ligados ao uso de uma interface por um usuário. Uma boa maneira de definir usabilidade, como definido em Lowdermilk[6](2013)

¹Project details; User requirements; Functional requirements; Database/dataflow diagrams; Screenshots of prototypes.

é: “o estudo de como humanos se relacionam com qualquer produto” (tradução livre)², enquanto a experiência de usuário é um conceito abrangente, que, como descrito em Albert[7](2013), necessita minimamente atender os seguintes critérios (tradução livre)³:

- Ter um usuário envolvido;
- Um usuário estar envolvido com a utilização do sistema, produto ou qualquer interface;
- A experiência deste usuário é de interesse, sendo possível observá-la e quantificá-la.

2.3 React

O React é uma tecnologia lançada pela companhia Facebook no ano de 2013, o que a faz uma tecnologia relativamente nova, considerando que existem diversas outras tecnologias de criação de Interface de Usuário (IU) estabelecidas há mais tempo. React é uma biblioteca de JavaScript (JS), uma linguagem de programação criada em meados da década de 1990, ou seja, o React é uma adição em cima da linguagem de programação no qual foi desenvolvido, facilitando o desenvolvimento que antes necessitaria ser feito com esforço maior e mais linhas de código, demandando mais tempo das equipes de desenvolvimento que utilizam esta linguagem.

A biblioteca React tem uma aceitação bem grande no mundo do desenvolvimento de software, especialmente nos últimos anos. Este fato se dá especialmente pela tecnologia ter sido criada e adotada pelo Facebook e, conseqüentemente, por suas subsidiárias, criando um sentimento de confiança na comunidade no que diz respeito à esta biblioteca. Exemplos de empresas grandes que utilizam essa biblioteca são: Netflix, Imgur, Airbnb, Twitter, Uber, Spotify, Instagram, Facebook e outras.

2.3.1 Por Que Usar React?

O React é uma biblioteca de código aberto (do inglês *open source*) de JavaScript. Isso significa que esta biblioteca possui licenciamento livre, ou seja, é permitido que usuários ao redor do mundo criem novas funcionalidades, modifiquem funcionalidades existentes e alterem o layout da biblioteca original sem ter que pagar nenhum tipo de licença. No geral, estas mudanças feitas por usuários são disponibilizadas como bibliotecas adicionais para o próprio React, mas, as vezes, as funcionalidades criadas pela comunidade são tão

²[...]is the study of how humans relate to any product.

³A user is involved; That user is interacting with a product, system or really anything with a interface; The user experience is of interest, and observable and measurable.

úteis ou necessárias (como é o caso de correção de erros encontrados por usuários), que são inseridas pelos mantenedores do React na biblioteca base.

Outro fator que favorece a escolha desta biblioteca às várias outras que existem no mercado é o fato de React permitir o uso de *Hypertext Markup Language (HTML)*- código que gera as páginas web- e *Cascading Style Sheets (CSS)*- código que gera a estilização das páginas web- dentro de seu código, facilitando o processo de aprendizado de programadores que não estão familiarizados com a linguagem JavaScript (JS), além de facilitar a implementação de diversas funcionalidades. Seguindo esta linha de raciocínio, é fácil implementar qualquer tipo de *script* já existente em outras páginas web, como por exemplo fazer a conexão do site com *feeds* do Instagram, Twitter e Youtube (caso haja necessidade de tal funcionalidade), já que os *scripts* utilizados em páginas web comuns são feitos em JavaScript.

Ainda sobre as vantagens do React, não se pode deixar de mencionar o amplo suporte disponível em diversas plataformas. Já foi mencionado anteriormente que a comunidade de React é grande e está sempre fazendo novas bibliotecas que ajudam no desenvolvimento de interfaces, mas, além disso, existem inúmeros fóruns para a grande maioria dos erros de sintaxe e erros de implementação que podem aparecer durante o processo de desenvolvimento, bem como uma boa documentação oficial (disponível em [8]) gerada pelo grupo de mantenedores. Grande parte dessa disponibilidade de suporte ocorre pela ampla aceitação da biblioteca, mas uma grande parte é devido ao fato da linguagem na qual é baseada já existir há tantos anos e ser uma das linguagens de programação mais utilizadas no mundo.

2.3.2 Componentes

Uma funcionalidade que faz o React tão popular entre os desenvolvedores é sua divisão de código em “blocos” chamados de componentes. O que isto quer dizer é que o código em React, caso seja bem escrito, pode ser extremamente eficiente em questão de redução de redundância de código. Para melhor descrever este conceito, pode-se usar o código padrão das páginas web: o HTML. Um site feito puramente em HTML (com adicionais como CSS e JS) que possui um cabeçalho ou um rodapé, deve ter o código deste mesmo cabeçalho e rodapé presente em todas as suas páginas, e este código deve ser duplicado toda vez que uma página nova for desenvolvida pelo programador. O React elimina esta necessidade explícita de código repetido, uma vez que basta criar um componente contendo este cabeçalho ou rodapé (como exemplificado) e adicionar estes às páginas que achar necessário.

2.3.3 Estados

O React possui um sistema de variáveis que controla suas páginas e componentes, chamados de estados (do inglês *state*). Estes estados podem ser entendidos como uma variável na página, podendo conter qualquer informação que o programador desejar armazenar no mesmo. A diferença do estado para uma variável comum é que o valor do estado não é atribuído simplesmente por meio de uma declaração do tipo:

```
1 estado = novoValor;
```

mas sim por meio da utilização de uma função que atribui um valor a um determinado estado.

A declaração do estado e de sua função de atribuição é feita utilizando uma funcionalidade nativa do React, chamada `useState`. Um exemplo de como declarar um estado e como alterar o mesmo é:

```
1 const [estado, setEstado] = useState(valorInicial);  
2 setEstado(novoValor);
```

Uma das maiores utilidades de um estado é a renderização⁴ de objetos (variáveis com várias propriedades internas), textos ou números, uma vez que o componente ao qual este estado pertence deve ser renderizado novamente sempre que um de seus estados tem o valor alterado, garantindo que o que está sendo mostrado na tela do usuário é de fato a informação mais recente que uma página possui.

2.3.4 Props

Outra funcionalidade interessante do React são os chamados *Props*. Este termo vem da palavra inglês referente a propriedades (*properties*). Os *props* servem o propósito de passar informações de um componente para o outro, geralmente de um componente pai para um componente filho (pai e filho sendo uma hierarquia na qual um componente contém o outro). Isto faz possível a “componentização” do React descrito na Seção 2.3.2, uma vez que os códigos referentes a dois componentes diferentes não estão presentes na mesma função, seria difícil passar informações entre eles.

Os *props* podem ter qualquer⁵ nome que o programador escolher, bastando explicitar este nome no momento de sua utilização no componente que o recebe. A seguir está exemplificada a declaração de um componente que recebe um *prop* chamado “nome” e o renderiza em uma forma de texto, bem como a maneira que o componente pai passa esta informação para o componente filho.

⁴Conceito utilizado para descrever o processo de gerar um elemento visual a partir de código.

⁵Contanto que não seja uma palavra reservada da linguagem de programação.

```

1 function ComponentePai() {
2     return (
3         <ComponenteFilho nome={'Nome do Usuario'} />
4     );
5 }
6
7 function ComponenteFilho({ nome }) {
8     return ( <p>{nome}</p> );
9 }

```

2.4 Leaflet

Para tornar possível a implementação de uma interface prática disponível em forma de mapa, foi utilizada a biblioteca de código chamada Leaflet. Esta biblioteca é responsável por maior parte das funcionalidades relacionadas à implementação da interface de mapa, uma vez que esta possui diversas funções relacionadas a tratamento de eventos (descritos em 2.4.1) de maneira nativa, ou seja, sem a necessidade de implementação adicional de código.

A biblioteca Leaflet foi escolhida por ser uma biblioteca de código aberto (assim como o React), sendo uma biblioteca com maior probabilidade de encontrar diversas implementações e soluções de problemas em fóruns na internet, bem como uma maior chance de ser constantemente atualizada pela comunidade com a autorização do autor original disponível em [9]. A biblioteca também conta com uma documentação extensa sobre todas suas funcionalidades e como utilizá-las no desenvolvimento de sistemas. Outro fator importante para a escolha desta é o fato de não necessitar de nenhuma chave de API (chave de acesso geralmente necessária para acessar serviços de empresas como Google e Facebook) e não necessitar nenhum tipo de pagamento para sua utilização completa. A boa performance desta biblioteca em sistemas web tanto em computadores quanto em celulares é outro fator levado em consideração na escolha de utilização do Leaflet para este sistema.

2.4.1 Eventos

Esta biblioteca, como mencionado anteriormente, faz de maneira nativa a implementação de diversas funcionalidades no que diz respeito a elementos em um mapa. Estas funcionalidades incluem criação, edição, exclusão e visualização de elementos geográficos (como marcadores de geolocalização), além de formas geométricas e traçados, como será descrito detalhadamente no Capítulo 4.

A própria biblioteca, ao criar, editar ou excluir um ou mais elementos, gera o que é chamado de evento. Por meio da utilização de *props* (assim como os descritos na Seção 2.3.4), podem ser atribuídas funções customizadas responsáveis pelo tratamento dos dados referentes aos elementos que estão presentes no mencionado evento, lançado pela biblioteca no momento que um determinado elemento é criado, editado ou excluído. Com esta funcionalidade nativa da biblioteca Leaflet, é possível implementar as funções desejadas para suprir as necessidades únicas do sistema de gestão de redes. Sem este tipo de customização no tratamento de eventos, o desenvolvimento da interface de mapa teria complexidade muito elevada e demandaria mais tempo da equipe de desenvolvimento, sendo assim uma funcionalidade de extrema importância.

2.5 Metodologia Ágil

É possível caracterizar conceito de metodologia ágil como sendo um conjunto de técnicas e práticas de gerenciamento de projetos- assim como descrito no manifesto ágil de Beck[10](2001)- que visa a praticidade, agilidade e qualidade do processo de desenvolvimento de software. Existem diversas formas de executar a metodologia ágil, existindo diversas ramificações da mesma, como, por exemplo: *Extreme Programming*, *Crystal Methods*, *Lean Development*, Scrum, dentre outras as quais não serão descritas em detalhes.

O ponto central da metodologia ágil é atingir uma gerência de projeto focada em entregar valor para o cliente com uma frequência alta, ou seja, focada em constantemente entregar pequenos pedaços do todo que é o software final. Para que isto aconteça, o método ágil faz o uso constante do *feedback*⁶ com o cliente, garantindo que todas as partes desenvolvidas do produto final são úteis e coerentes com o modelo de negócio desenvolvido pelo cliente, além de terem a qualidade desejada. Outra característica que destaca esta metodologia é o fato de ser iterativa, o que significa que o projeto é dividido em etapas que se repetem periodicamente ao longo de todo o processo de desenvolvimento.

2.5.1 Scrum

A metodologia ágil utilizada para o desenvolvimento do sistema de gestão de redes foi a metodologia Scrum descrito em Schwaber[11](2002). Esta consiste na subdivisão do projeto como um todo em pequenas funcionalidades, que são inicialmente categorizadas como *backlog*, uma lista de funcionalidades a serem implementadas que devem ser entregues ao cliente ao final de um período pré determinado chamado de *sprint*.

⁶Termo do inglês que pode ser definido como o processo de mostrar um resultado e receber críticas sobre o mesmo.

O cliente de um projeto é a pessoa, ou grupo de pessoas, que contrata o serviço de desenvolvimento de um determinado produto, seja este um sistema, um aplicativo ou outros. Além do cliente existe o usuário deste determinado sistema, que pode ou não ser o cliente. No sistema descrito neste documento, o cliente em questão é o presidente do comitê técnico da GigaCandanga e o usuário deste sistema é o supervisor de infraestrutura, ambos citados na seção 1.1.

Sprint

Sprint é o nome dado ao período de desenvolvimento na metodologia SCRUM. O tempo que uma *sprint* pode durar é baseado no nível de dificuldade das funcionalidades desejadas para este período, o que geralmente tem duração de duas a quatro semanas. Esta técnica de separação e entrega de funcionalidades é amplamente adotada no mercado de desenvolvimento de software por trazer resultados frequentes e de qualidade superior, uma vez que o time de desenvolvimento pode focar em garantir o pleno funcionamento do que está sendo desenvolvido, ao contrário de outras técnicas, nas quais os desenvolvedores apressam o desenvolvimento do sistema como um todo e acabam gerando um projeto repleto de *bugs*⁷.

Histórias de Usuário

Por fim apresenta-se o conceito de histórias de usuário: são funcionalidades vistas de maneira macro, ou seja, menos subdivididas, que servem para contextualizar o que deverá ser desenvolvido ao decorrer de cada *sprint*. Um exemplo fictício de história de usuário é a funcionalidade de colorir todos os elementos que aparecem no mapa. Para isto, devem ser feitas várias subdivisões de tarefas, como, por exemplo: colorir todas as caixas de emenda; colorir todas as instituições parceiras; colorir todos os trechos de cabo e assim por diante.

Daily Meeting

A prática de *Daily Meeting*- traduzido para “reunião diária”⁸ consiste em realizar reuniões diárias curtas com toda a equipe de desenvolvimento, para que todos possam falar sobre o que fizeram após a reunião do dia anterior até a reunião do dia em questão, bem como quaisquer problemas que possam ter encontrado ao longo deste período de desenvolvimento, além de falarem se algo os impede de prosseguir com o desenvolvimento do sistema, para que seus colegas possam os ajudar.

⁷Termo utilizado para se referir à problemas inesperados no código.

⁸Tradução livre.

2.6 Resumo Conclusivo

Neste capítulo foram explicados os conceitos que serão utilizados ao longo deste trabalho. O primeiro conceito explicado foi o *Design* Centrado no Usuário, um conceito que pode ser descrito como o design de interfaces baseado em um processo constante de realização de testes de utilização, *feedback* do usuário e levantamento de requisitos a partir destes. A aplicação destas estratégias de *design* foi de grande importância para o desenvolvimento do sistema de gestão de redes. Em seguida foram explicadas as tecnologias utilizadas no desenvolvimento do sistema, tal como os motivos para estas tecnologias terem sido escolhidas. A utilização do React como biblioteca de desenvolvimento de interface em JavaScript foi baseada na facilidade de uso desta, além de outros fatores como a ampla aceitação do React no meio de desenvolvimento de interfaces. A biblioteca de mapa Leaflet foi escolhida por sua ampla implementação em diversos sistemas, visando facilitar a implementação de funcionalidades existentes em outros sistemas similares ao sistema de gestão de redes. Além disto, a biblioteca Leaflet é uma biblioteca gratuita que não necessita de chaves de API. Por fim, foi explicado o conceito de SCRUM, um método ágil que consiste em entregas periódicas de funcionalidades para o cliente, garantindo um bom fluxo de desenvolvimento do sistema, além de uma melhor organização do processo de desenvolvimento em si.

Este capítulo é de grande importância para o entendimento deste trabalho, visto que a metodologia empregada no desenvolvimento da interface do sistema é baseada nos conceitos nele explicados. A partir destas definições é possível seguir para o Capítulo 3, no qual é descrita a metodologia utilizada para o desenvolvimento do sistema tratado neste trabalho, além de ser possível entender a aplicação de alguns dos conceitos aqui explicados.

Capítulo 3

Metodologia

Para um desenvolvimento eficiente e que segue as regras de Design Centrado no Usuário e as práticas do desenvolvimento ágil, foi necessário estabelecer uma metodologia a ser seguida. Este processo envolve a tomada de diversas decisões de projeto, desde a maneira como cada página da interface deve ser desenvolvida, até como o processo de *feedback* do usuário deve ser feito. Este capítulo visa explicar como foi este processo de planejamento e execução (à nível estrutural) do desenvolvimento do sistema.

3.1 Definição de Objetivos

O primeiro passo tomado para executar o processo de planejamento do sistema foi entender as expectativas do usuário para o sistema final referente a funcionalidades macro¹. Para que este processo fosse executado da melhor maneira possível, o início do projeto foi marcado por diversas reuniões para entender de qual maneira a GigaCandanga fazia a gestão de sua rede de fibras sem o uso do sistema encomendado.

Durante as primeiras semanas do projeto, foram feitas reuniões entre a equipe de desenvolvimento, *front-end* e *back-end*, e a equipe técnica da GigaCandanga, para que todos os conceitos de elementos que compõem a rede fossem devidamente entendidos por todos os desenvolvedores. Após terem sido apresentados todos os componentes descritos na Seção 2.1 (de maneira detalhada), a equipe de infraestrutura apresentou para a equipe de desenvolvimento os softwares utilizados para a gestão da rede. Um desses softwares era responsável pela visualização da rede em mapa, para o controle geográfico dos componentes ali presentes, bem como estatísticas sobre estes componentes. O outro software utilizado era o Microsoft Excel [12], um programa de utilização de planilhas, onde estavam presentes todos os dados referentes aos componentes que compõem a rede, bem como as informações que conectam cada elemento da mesma, por exemplo: identificadores únicos

¹Funcionalidades mais abstratas, sem um nível de detalhamento tão grande.

(chamados de chaves ou *ids*) de uma caixa de emenda na tabela referente aos segmentos, especificando que esta caixa faz parte deste segmento, bem como outros atributos importantes para a identificação de cada elemento.

3.1.1 Funcionalidades

Concluída a etapa anterior, foi possível entender as funcionalidades que o sistema encomendado deveria ser capaz de prover. Além das reuniões mencionadas anteriormente e da apresentação dos métodos de gerenciamentos utilizados pela empresa, foi apresentado à equipe de desenvolvimento um esboço da estrutura lógica de um banco de dados que havia sido feita em outro momento por uma diferente equipe de desenvolvimento que não levou o projeto à diante. Possuir este esboço permitiu que a equipe pudesse fazer, juntamente com o usuário e o cliente, um levantamento geral de funcionalidades que o sistema precisaria ter para o funcionamento desejado.

Cada tipo de elemento da rede possui diferenças em relação aos demais elementos, gerando uma complexidade no que se refere às relações entre elementos. Um exemplo destas diferenças é: uma caixa de emenda precisa ter um trecho de cabo associado, enquanto um poste precisa ter uma caixa de emenda e pode ter múltiplos trechos de cabo associados à ele. Ao longo da primeira etapa de desenvolvimento foram feitas diversas alterações ao modelo do banco de dados exemplificado na Figura ?? sendo que algumas mudanças foram feitas pela equipe do *back-end* quando o sistema já estava avançado no processo de desenvolvimento.

3.1.2 Requisitos do Usuário

Uma vez que todos os membros do time de desenvolvimento já tinham em mente como a gestão das redes era feita previamente ao desenvolvimento do novo sistema, tornou-se então necessário estabelecer os requisitos que o usuário final tinha como mais necessários e mais desejados. Perceba que existe uma diferença entre as funcionalidades que o sistema deve ter para seu funcionamento geral e o que o usuário enxerga ser necessário. Um exemplo desta diferença pode ser dado pela analogia de um sistema de caixa de supermercados: uma funcionalidade que este sistema deve ter é a possibilidade de adicionar múltiplos de um mesmo item, o que pode ser feito passando o mesmo produto múltiplas vezes e o sistema agrega todas estas entradas como múltiplos de um determinado produto; já uma necessidade para o atendente do caixa, é que exista um campo onde pode ser digitado em uma única vez a quantidade desejada de um determinado produto, havendo a necessidade de registrar o código do produto apenas uma vez. A partir desta diferença

de conceitos, pode-se fazer o levantamento de funcionalidades que o usuário deseja que o sistema encomendado possua para facilitar sua usabilidade.

O processo de coleta de requisitos foi feito diretamente com o usuário mais envolvido com o uso deste sistema, uma vez que ele é o usuário que mais entende das dificuldades e pontos a serem melhorados da gestão prévia. A equipe de *front-end* analisou as planilhas utilizadas pelo usuário para entender quais dados e conexões eram necessários para cada elemento, bem como teve diversas reuniões com o usuário sobre a necessidade de cada tipo de dado. Com isto, ficou mais claro como o fluxo do sistema deveria ser feito. Apesar disto, ainda sim diversos aspectos desse fluxo sofreram mudanças, uma vez que ao longo do desenvolvimento novas necessidades são encontradas ou requisitadas pelo próprio usuário.

3.2 Prototipação

Com o levantamento inicial de necessidades e requisitos, a equipe do *front-end* pôde então dar início ao processo de prototipação das interfaces onde o CRUD de cada elemento individual é feito. A prototipação desta parte da interface web foi feita no software Figma [13], e foi apresentada para o usuário para que o mesmo pudesse dar o *feedback* necessário para o melhoramento da interface, garantindo que suas necessidades sejam atendidas. A Figura 3.1 e Figura 3.2 representam o protótipo de uma das páginas de visualização de dados em tabela, porém, com campos de dados não reais, uma vez que nesta fase do projeto ainda não eram certos quais campos seriam estritamente necessários para este tipo de elemento. Na Figura 3.1, o ícone à esquerda dos dados da tabela- referente à funcionalidade de ver anotações sobre o elemento em questão- e o ícone contendo uma seta circular presente ao lado direito dos dados- que se refere ao histórico do elemento em questão- estão presentes no protótipo mas não foram implementados na versão final desta interface por não terem sido julgados relevantes para esta versão do sistema. Além disso é importante ressaltar que neste protótipo foi apresentada a funcionalidade de exclusão de elementos em lote, a pedido do usuário.

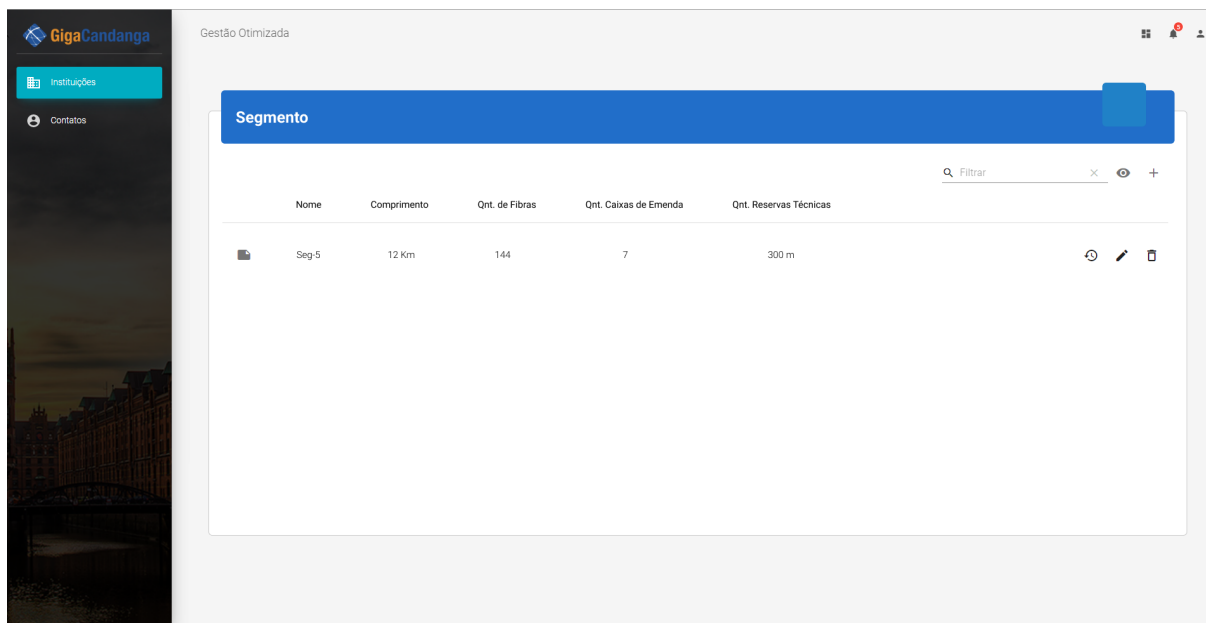


Figura 3.1: Protótipo da página de tabela de segmentos, com opção de criar, editar, e excluir segmentos.

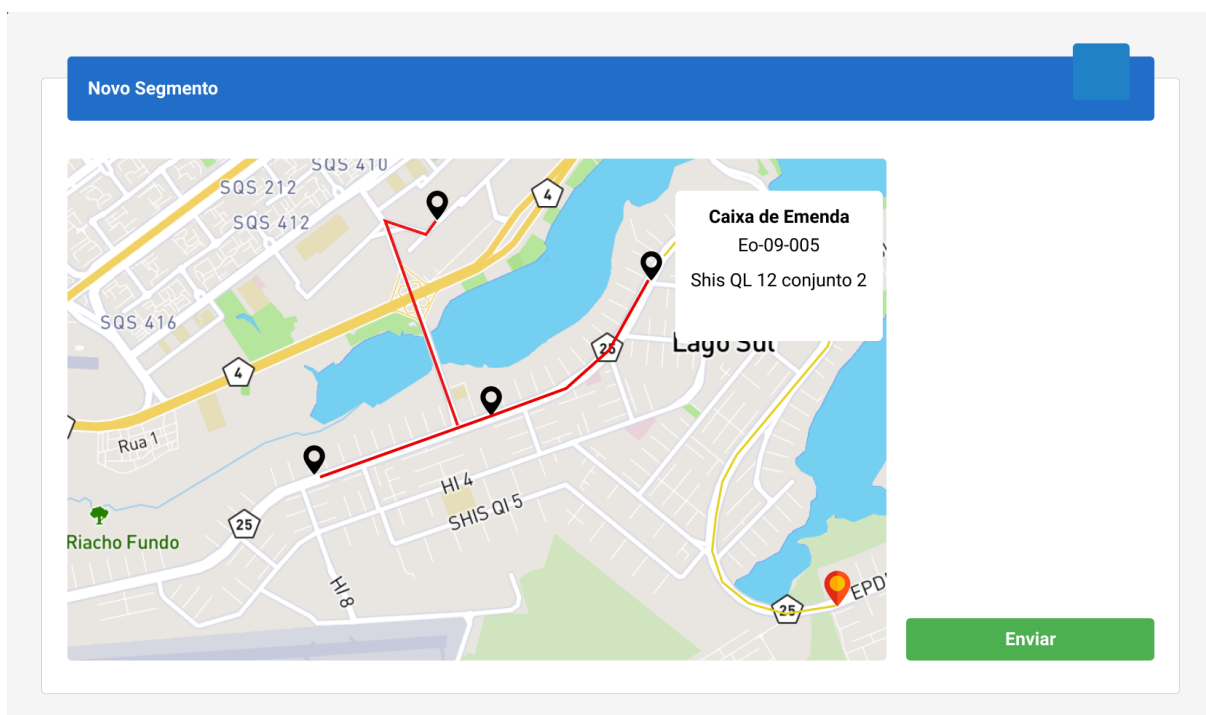


Figura 3.2: Protótipo do *pop-up* de mapa utilizado para escolher localização dos elementos.

O processo de prototipação foi feito na etapa inicial do desenvolvimento, antes mesmo da equipe dar início à escrita de código referente ao sistema, uma vez que a grande maioria

das páginas que compõem o sistema possuem o *layout* muito semelhante, permitindo a reutilização do protótipo inicial.

3.3 Execução das Sprints

Como explicado na Seção 2.5.1, o desenvolvimento foi dividido em ciclos de entrega de funcionalidades, chamados de *sprints*. O tempo que cada ciclo dura é relativo à complexidade das funcionalidades e mudanças implementadas no código, mas é possível traçar uma média de tempo que estas *sprints* duraram para um período entre duas e quatro semanas, sendo quatro semanas uma *sprint* de complexidade elevada. A definição de quais funcionalidades e mudanças seriam implementadas em cada *sprint* foi feita pelo gerente de projeto contratado pela GigaCandanga. O gerente de projetos, juntamente com o usuário, definia quais eram as prioridades à serem desenvolvidas, baseadas tanto na necessidade imediata do usuário final quanto do fluxo de desenvolvimento que já estava sendo seguido. O time de desenvolvimento também tem um lugar relevante na escolha destas funcionalidades, uma vez que os mesmos podem falar para o gerente de projeto o fluxo de desenvolvimento que julgam ser mais válido.

3.3.1 Demonstração das funcionalidades

Ao final de cada período estipulado, foi realizada uma reunião de finalização de *sprint*, na qual o usuário, o gerente de projeto e os membros dos times de desenvolvimento estavam presentes. Nestas reuniões cada equipe fala sobre as funcionalidades implementadas, possíveis problemas que podem ter encontrado ao longo do desenvolvimento e também, caso seja relevante (o *back-end* nem sempre consegue demonstrar o que é feito já que seus resultados não necessariamente têm interface), demonstram as novas funcionalidades do sistema e podem, assim, coletar *feedbacks* do usuário para melhorar o que o foi entregue nesta *sprint*.

3.4 Resumo Conclusivo

Este capítulo descreveu como foi definida a metodologia do desenvolvimento do sistema. A primeira etapa do desenvolvimento consistiu em definir os objetivos que o cliente e o usuário têm para o funcionamento final do sistema, fazendo o levantamento de requisitos e funcionalidades. A partir disto, foi dado início à etapa de prototipação da interface, por meio do uso de um software de prototipação. O protótipo inicial foi importante para auxiliar o usuário na definição de funcionalidades mais específicas que cada parte da

interface deveria ser capaz de executar. A execução do desenvolvimento foi então dividido em *sprints*, de acordo com as práticas do desenvolvimento ágil e *scrum*, como explicado na Seção 2.5.1. Por fim, foi descrito o processo de demonstração de funcionalidades para o usuário, garantindo que o mesmo pudesse fazer críticas relacionadas à versão apresentada.

A partir do entendimento de como a metodologia definida afeta o fluxo do desenvolvimento do sistema, é possível seguir para a etapa de implementação das funcionalidades, na qual é descrito em detalhes o processo de codificação do sistema e sua interface.

Capítulo 4

Implementação

A partir da execução da fase de planejamento e listagem de requisitos do sistema, é possível avançar para a próxima etapa do projeto: a execução. Este capítulo descreverá o processo de desenvolvimento do sistema, falando sobre a escrita de código em si e também descrevendo funcionamentos e escolhas feitas ao longo do trajeto.

4.1 *Pop-ups*

O termo *pop-up* é um termo em inglês referente às pequenas janelas que aparecem, geralmente, em interfaces web. A utilidade de um *pop-up* é mostrar informações adicionais sem redirecionar o usuário para outra página dentro de um site, e foi justamente para esta função que foi utilizado no contexto deste sistema. Outro termo utilizado para este tipo de janela é “modal”.

Para este projeto, os *pop-ups* foram utilizados para lidar com inserção de dados referentes aos elementos que compõem a rede, ou seja, foram utilizados para lidar com formulários sem haver a necessidade de redirecionar o usuário para outra página dentro do sistema. Esta necessidade foi considerada essencial, visto que a página de mapa centraliza em um ponto pré definido sempre que é recarregada, o que faria a experiência do usuário ser afetada tendo que movimentar o mapa a fim de centralizar o ponto ou região que estava observando anteriormente.

4.1.1 Criação

Os *pop-ups* são utilizados para lidar com o formulário de criação de todos os elementos descritos em 2.1, e também é utilizado para grande parte das edições de informações à respeito destes elementos, com algumas exceções que serão descritas posteriormente. A

Figura 4.1 apresenta um exemplo de um *pop-up* de criação de *jumper*, e a Figura 4.2 apresenta um exemplo de um modal de criação de caixa de emenda:

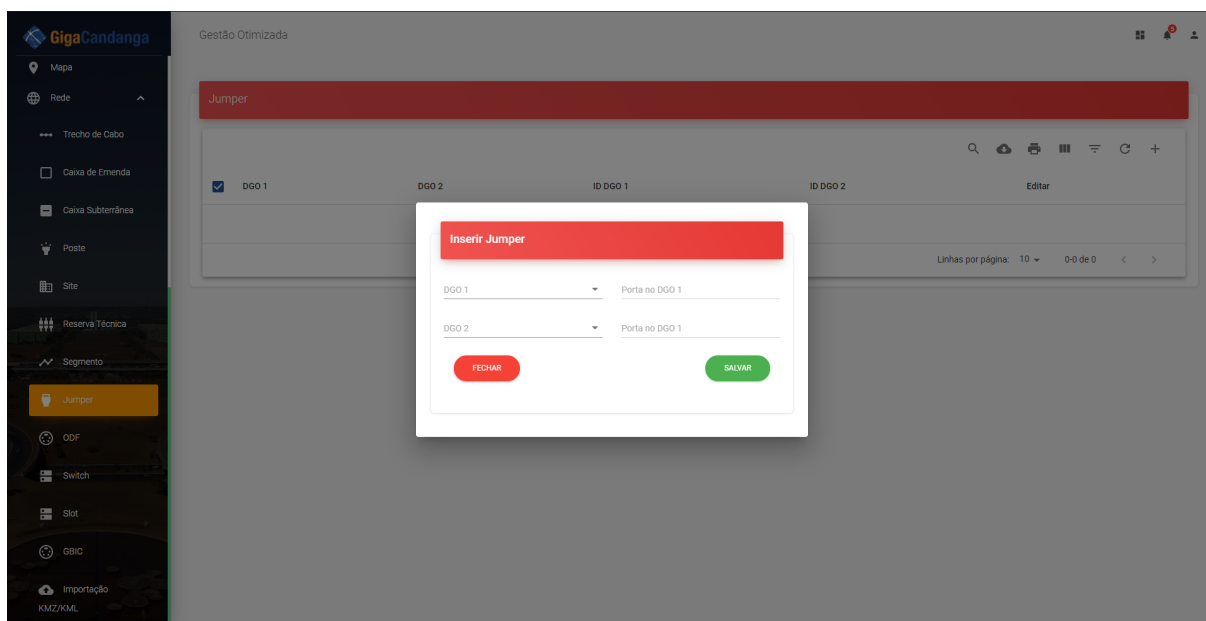


Figura 4.1: *Pop-up* de criação de *jumper*. Este formulário contém seletores para fazer a associação com outros elementos da rede.

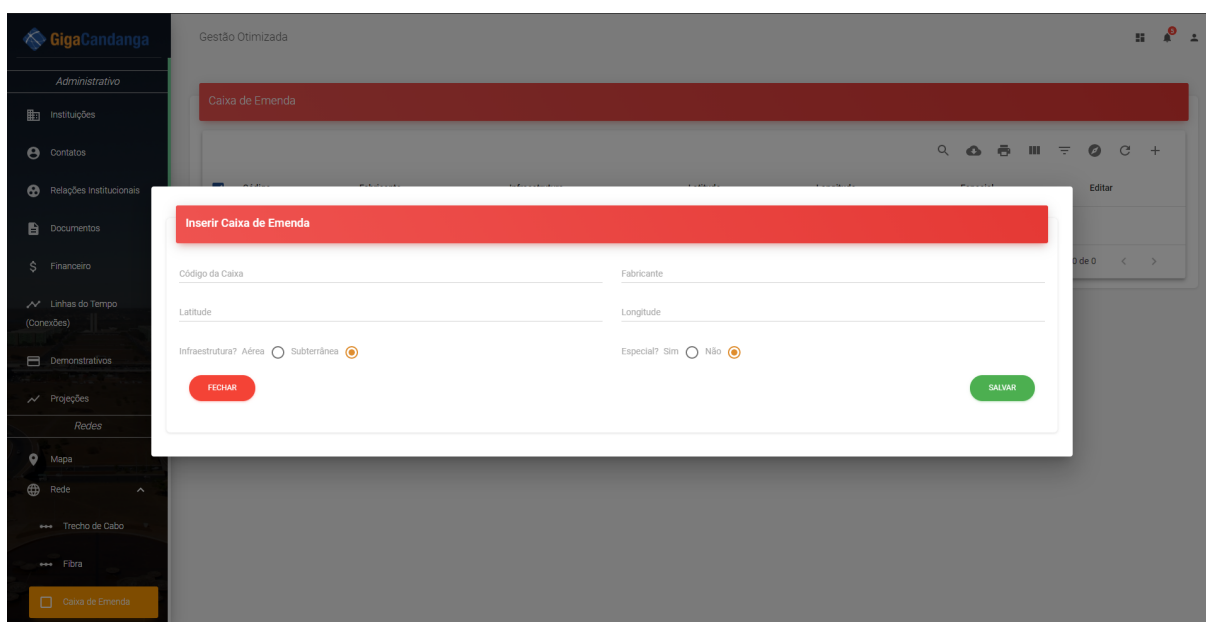


Figura 4.2: *Pop-up* de criação de caixa de emenda. Este formulário contém campos numéricos e de texto.

4.1.2 Edição

A funcionalidade de edição de dados por *pop-up* funciona de maneira semelhante à criação, com a diferença sendo as *props* passadas para o componente do formulário. Estas *props* são uma variável booleana¹ e uma variável contendo o objeto à ser editado. A *prop* booleana é chamada “*edit*”, e é responsável pela diferenciação do tratamento de dados pelo componente, bem como alterar visualmente o *pop-up*, como é o caso do título deste, que passa de “Inserir Elemento” para “Editar Elemento”. A outra *prop* tem seu nome variado, mas sempre corresponde ao nome do elemento da rede que está sendo manipulado, e contém todos os campos que este elemento inicialmente possui, como nome, número de registro, comprimento (para o exemplo de cabos), dentre outros. Com estes dados passados para o formulário por meio das *props*, é possível executar a renderização do *pop-up* na tela do usuário com seus dados do elementos já preenchidos.

4.1.3 Outras Funcionalidades

Seleção de Elementos Associados

Muitos elementos necessitam de uma atribuição à outros elementos distintos da rede. Isto pode ser feito de duas maneiras: atribuição por campo seletor para os elementos que possuem nomes distintos, como é o caso na Figura 4.1, onde um *jumper* precisa ser associado a um DGO; e seleção de elementos que não possuem nome, como é o caso de postes e fibras, que necessitam atribuir trecho(s) de cabo a seu cadastro por meio de um *pop-up* com um mapa parcial (ocupa uma pequena porção da tela) para visualizar e selecionar os elementos a serem associados, como exemplificado na Figura 4.3.

¹Variável que possui apenas dois possíveis valores: verdadeiro ou falso

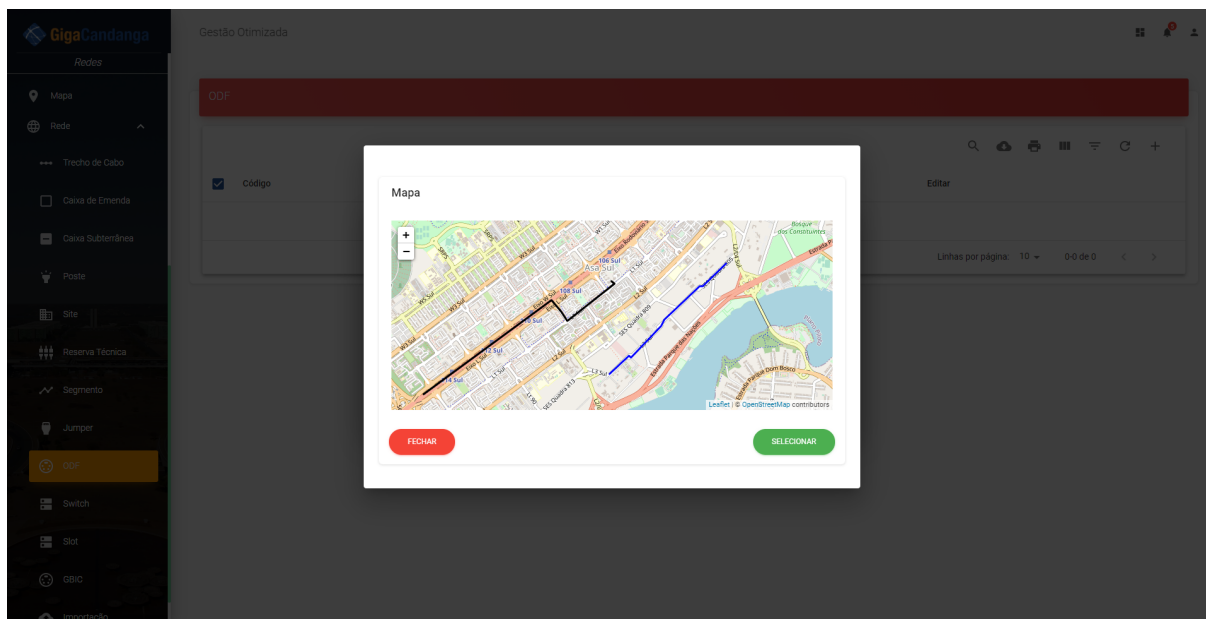


Figura 4.3: *Pop-up* de seleção de trechos de cabo à serem associados. Linhas azuis representam trechos de cabo selecionados pelo usuário.

Filtragem

A última funcionalidade diferenciada dos *pop-ups* é sua utilização para filtragem nas páginas de visualização por tabelas- descritas na Seção 4.2. Os *pop-ups* são utilizados nestas páginas para mostrar um mapa- sem nenhum elemento- que serve o propósito de selecionar uma área de visualização do mapa para qual o usuário quer filtrar os elementos, ou seja, ao selecionar uma área visível no mapa, apenas os elementos que estão dentro desta região serão apresentados na tabela.

4.2 Tabelas

Para cada elemento da rede foi desenvolvida uma página para a execução do CRUD de seus dados em forma de tabela. Estas tabelas foram implementadas utilizando uma biblioteca de código, ou seja, é um código de terceiros que recebe parâmetros e os renderiza em forma de tabela.

A tabela presente em cada uma destas páginas tem a capacidade nativa, ou seja, já implementada na biblioteca original, de filtrar os elementos baseado em valores de seus campos (escolhidos no momento da implementação no código); ordenar elementos baseado em valores de suas colunas; buscar elementos baseado no conteúdo de seus campos e excluir múltiplos elementos- esta é uma funcionalidade mista, uma vez que a biblioteca de tabelas não tem acesso ao banco de dados do sistema, sendo necessário integrar estas

funcionalidades manualmente. Nas seguintes subseções serão explicadas e exemplificadas as funcionalidades que foram implementadas manualmente, pelo fato da biblioteca não possuir estas funcionalidades, ou de não atender as necessidades específicas que o sistema demanda destas.

4.2.1 Filtragem por Geolocalização

Uma funcionalidade que houve necessidade de ser implementada de maneira manual é a filtragem por localização. A funcionalidade de filtragem já existe na tabela sem a necessidade de ter que implementar nada a mais, porém, o diferencial desta funcionalidade é a existência de dois campos responsáveis por localização que são os campos de latitude e longitude. O primeiro problema que surge com esta disposição de dados é o fato do usuário necessitar filtrar duas colunas para devidamente filtrar os elementos de uma tabela por sua localização, fazendo este tipo de filtragem inviável para o usuário final, bem como extremamente não amigável ao uso por este usuário.

A funcionalidade em questão foi implementada para suprir a necessidade de filtrar por uma região de um mapa. Utilizando um *pop-up* com um mapa de tamanho reduzido, como descrito em 4.1.3, o sistema permite que o usuário arraste o mapa para a localização desejada, permitindo também aumentar ou diminuir o zoom do mapa para ter o controle preciso da região na qual deseja visualizar os elementos. A Figura 4.4 apresenta o ícone utilizado para a funcionalidade de filtragem por geolocalização:



Figura 4.4: Ícone associado à filtragem por geolocalização (Fonte: [14]).

4.2.2 Adição de Elementos

Como explicado na Seção 4.1.1, para a adição de novos dados às tabelas de cada elemento, foram utilizados os diferentes tipos de *pop-ups*. Para elementos que necessitam selecionar trechos de cabo ou outras entidades que não possuem nome ou que possuem nome não distintivo, o processo de cadastro é: primeiro surge o *pop-up* principal, contendo os campos referentes à um elemento que não necessitam de nenhuma adição via mapa, e, depois que

o usuário salvar os dados preenchidos, ao clicar no botão de seleção de elementos, o *pop-up* de mapa automaticamente é aberto, mantendo as informações anteriores salvas em um estado do React. Ao terminar a seleção dos elementos desejados, o formulário completamente preenchido pelo usuário é enviado ao servidor e os devidos elementos são criados no banco de dados, automaticamente atualizando a tabela com os novos dados.

4.2.3 Edição de Elementos

Para a edição de elementos presentes nas tabelas o processo é similar ao descrito na Seção 4.2.2, com a diferença das *props* passadas para o componente do formulário, como descrito em 4.1.2.

4.3 Mapa

Tendo descrito a utilização e o processo de manipulação de dados dos elementos da rede em suas páginas dedicadas, pode-se passar a funcionalidade vista como a mais essencial da interface do sistema de gerenciamento de redes: o mapa. A página de mapa é considerada a página mais relevante pois é nesta página que a visualização da rede acontece de uma maneira generalizada, com seus elementos dispostos geograficamente sobre o mapa da cidade, diferentemente das páginas dedicadas de cada elemento, onde o usuário apenas visualiza o elemento correspondente à página que se encontra e os dados destes elementos são dispostos somente em forma de tabelas, dificultando a compreensão da rede como um todo.

A Figura 4.5 apresenta a interface do mapa descrita ao longo da Seção 4.3. No canto superior esquerdo está localizada a barra de ferramentas, na qual os seis primeiros itens de cima para baixo são nativos da biblioteca Leaflet, e os demais são seletores de tipo de elemento, assim como descrito na Seção 4.3.2. Todos os ícones, tanto nativos quanto customizados, possuem *tooltips* descrevendo a função que cada ícone realiza. No canto superior direito está localizado o seletor de layout de mapa descrito na Seção 4.3.1.



Figura 4.5: Visão geral da interface de mapa. Os ícones ao lado esquerdo da imagem são funcionalidades de manipulação do mapa e de elementos nele mostrados. No canto superior direito está presente o menu seletor de *layout* de mapa. No centro do mapa- em cor preta- estão representados trechos de cabo, caixas de emenda e postes.

4.3.1 Visualização de Elementos

A visualização de elementos por meio do mapa consiste na utilização da biblioteca Leaflet, mencionada em 2.4. Esta biblioteca faz, de maneira nativa, a renderização de marcadores geográficos próprios, bem como a renderização de linhas poligonais (chamadas *polylines*), balões de informação e outros elementos que foram desabilitados por terem sido julgados desnecessários para as funcionalidades deste sistema, como círculos e outras formas geométricas.

A biblioteca Leaflet permite o programador ter a opção de customizar o ícone utilizado para a renderização de seus marcadores geográficos. Esta customização foi utilizada para diferenciar os ícones dos diferentes tipos de elementos da rede que são disponibilizados no mapa. Além disto a biblioteca permite que o desenvolvedor, ao renderizar os marcadores geográficos, passe *props* customizadas para os mesmos, o que foi extremamente necessário tanto para a manipulação destes elementos (descrita nas Seções 4.3.2, 4.3.3 e 4.3.4), quanto para a visualização das informações atreladas à estes por meio de seus balões de

informação.

Para a visualização de trechos de cabo, foi necessário utilizar a funcionalidade de renderização em mapa dos elementos previamente mencionados chamados *polylines*. O que diferencia estas linhas poligonais de segmentos de retas é a maneira como estas são formadas. As *polylines* são formadas por diversos pontos, que, por sua vez, são conectados por segmentos de retas. Com isto, é possível obter trechos de cabos renderizados através do mapa de maneira extremamente fiel à sua posição real.

Para a melhor usabilidade (à pedido do usuário) e melhor visualização dos elementos no mapa, foi implementada a funcionalidade de múltiplos layouts de mapa. Esta funcionalidade é nativa da biblioteca Leaflet, facilitando sua implementação. A Figura 4.6 e Figura 4.7 exemplificam a visualização do mapa em diferentes layouts. É importante ressaltar que para utilizar as funcionalidades nos diferentes layouts de mapa, além de remover as marcas d'água presentes é necessário adicionar um método de pagamento para gerar uma chave de API do Google.

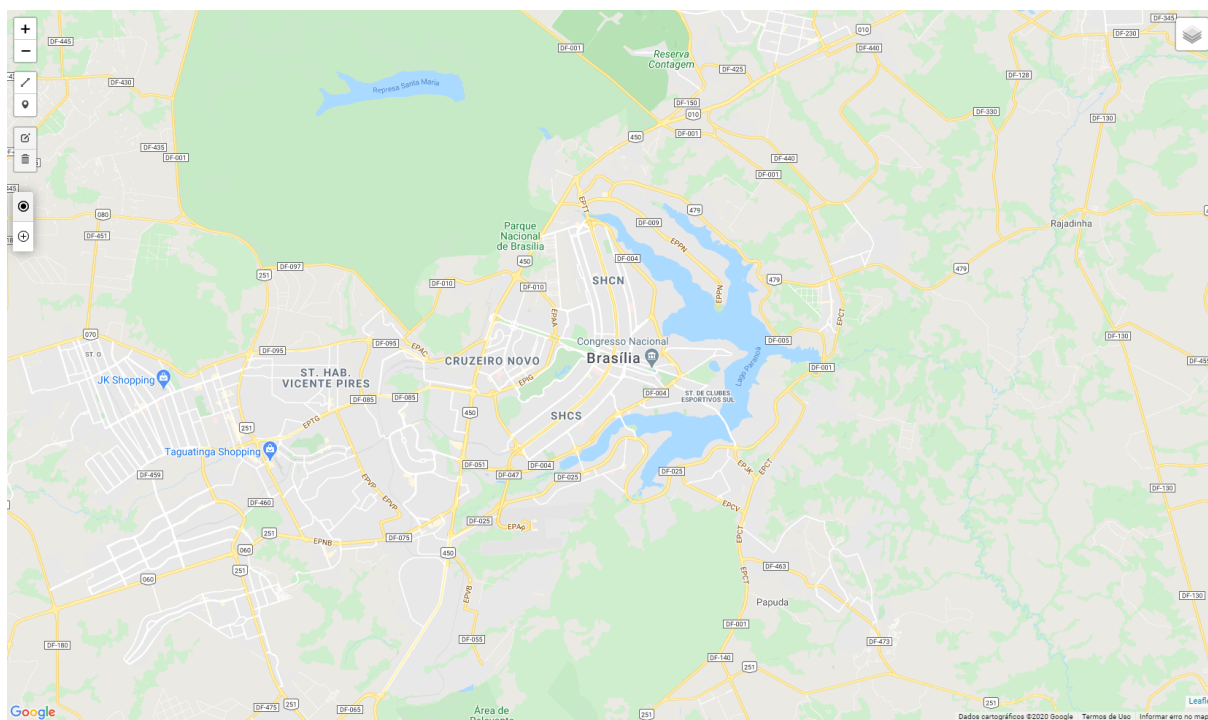


Figura 4.6: Layout do mapa utilizando a API do Google Maps Roads.

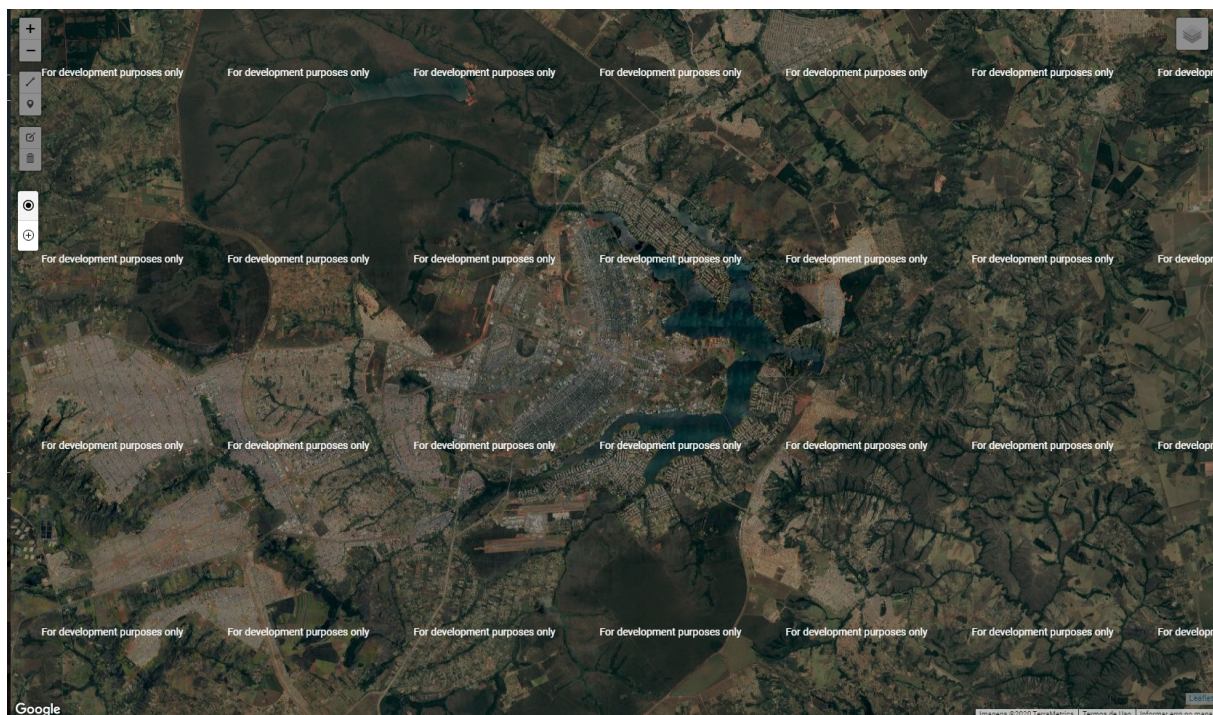


Figura 4.7: Layout do mapa utilizando a API do Google Maps Satellite.

4.3.2 Criação de Elementos

A maior diferença do modelo de criação de elementos via interface de mapa em comparação com a interface de tabela é a possibilidade de utilizar as funcionalidades da biblioteca Leaflet. Estas funcionalidades permitem o usuário criar elementos baseados em sua posição no mapa, sem haver a necessidade do usuário saber a latitude e a longitude física dos elementos, bastando apenas identificar a posição na qual o elemento existe fisicamente.

Para criar elementos representados por um marcador geográfico (qualquer elemento que não seja trecho de cabo e que seja permitido criar por esta página) foi utilizada a funcionalidade nativa da biblioteca, customizando os ícones para cada tipo de elemento, assim como explicado na Seção 4.3.1. Como a biblioteca não prevê o uso de diversos tipos de marcadores, foi necessário criar um componente próprio ao sistema, capaz de selecionar qual tipo de elemento o usuário deseja criar. Ao clicar em um tipo de elemento, é armazenado em um estado qual o tipo selecionado, utilizando esta informação no momento que a biblioteca dispara o evento de criação descrito na Seção 2.4.1. No momento que este evento de criação é disparado, o sistema automaticamente abre um *pop-up* com os dados de latitude e longitude já preenchidos, permitindo o usuário editar estes dados, bem como preencher os outros dados que estão faltando.

A criação de trechos de cabo ocorre de maneira semelhante. A diferença deste tipo de elemento para os outros é que o trecho de cabo necessita ser “desenhado” sobre o mapa pelo

usuário, utilizando a funcionalidade nativa de desenhar *polylines*. Ao finalizar o desenho sobre o mapa, *pop-up* de criação de elemento aparece para o usuário, permitindo que o mesmo complete o cadastro com as informações desejadas.

4.3.3 Edição de Elementos

A funcionalidade de edição de elementos no mapa também é uma funcionalidade nativa da biblioteca Leaflet. Na barra de ferramentas nativa da biblioteca existe uma opção que, ao ser clicada, permite a edição de todos os elementos que estão renderizados dentro do componente do mapa. Ao finalizar a edição, por meio do botão também presente na barra de ferramentas, o componente do mapa dispara um evento de edição, onde é tratado cada tipo de componente que foi editado, utilizando o *prop* passado para o marcador, como mencionado na Seção 4.3.1.

A edição de trechos de cabo pode ser feita de maneira extremamente precisa, uma vez que pode ser editado cada ponto que compõe a linha poligonal, bem como criar novos pontos permitindo o usuário deixar o trecho de cabo ainda mais preciso em relação à sua posição real. A edição, por meio da interface de mapa, deste tipo de elemento afeta somente o trajeto pelo qual o trecho de cabo passa, não mostrando o *pop-up* de edição para o usuário alterar outros campos.

O processo de edição de elementos que são representados por marcadores geográficos é semelhante ao processo de edição de trechos de cabo, com uma diferença: os elementos que necessitam escolher outros elementos diversos para fazer uma associação (como é o caso para o poste) têm a possibilidade de editar outros campos que não apenas a latitude e a longitude, contanto que somente um elemento seja editado por vez. Esta decisão foi feita visando garantir maior fluidez à utilização do sistema, bloqueando o aparecimento do *pop-up* de edição caso vários elementos sejam editados de uma única vez, evitando que o usuário gaste tempo desnecessário fechando todas as janelas adicionais que apareceriam caso todos os elementos renderizassem seu formulário de edição.

4.3.4 Exclusão de Elementos

Assim como as funcionalidades de criação e edição, a exclusão de elementos é feita a partir da funcionalidade nativa da biblioteca Leaflet. Na barra de ferramentas o usuário pode escolher a ferramenta de exclusão, permitindo o mesmo selecionar todos os elementos que deseja excluir. Ao finalizar, o usuário deve apertar uma confirmação, evitando a exclusão acidental. Ao finalizar este processo, o componente do mapa dispara o evento de exclusão que é tratado utilizando o *prop* mencionado na Seção 4.3.1 e enviado para o servidor, devidamente removendo estes elementos do banco de dados.

4.4 Resumo Conclusivo

Este capítulo detalhou o processo de implementação das funcionalidades do sistema de gestão de redes, explicando as decisões tomadas para cada tipo de utilidade das interfaces mencionadas. A primeira funcionalidade descrita foi a utilização de *pop-ups* para implementar formulários de cadastro e edição, bem como outras funcionalidades diversificadas como a filtragem por geolocalização e seleção de elementos associados. Em seguida foi explicado o processo de implementação das páginas com visualização em forma de tabela, utilizadas pelo usuário para facilitar a edição de elementos específicos, bem como a manipulação de dados sem haver a necessidade de interagir com a interface de mapa. Por último foi descrito o processo de implementação da interface de mapa, capaz de realizar edição, exclusão e criação de elementos, além da visualização destes. Este capítulo é relevante para o entender o funcionamento desta versão do sistema, visto que o usuário realizou testes baseados nas funcionalidades descritas neste capítulo.

Entendendo o funcionamento geral do sistema, é possível então avançar no desenvolvimento deste trabalho, partindo para os resultados obtidos com os testes realizados pelo usuário final deste sistema.

Capítulo 5

Resposta dos Usuários

O objetivo deste capítulo é descrever o processo de avaliação do sistema por seu usuário e então calcular, a partir de métricas pré-estabelecidas, a experiência deste usuário. Serão explicadas as métricas utilizadas para a quantificação da experiência, bem como o reflexo destes resultados no estado geral do desenvolvimento do sistema.

5.1 Métricas de Experiência do Usuário

A experiência do usuário, como já explicada em 2.2.1, é um fenômeno quantificável que surge a partir da interação de um usuário com uma interface. Existem diversas métricas utilizadas para quantificar esta experiência em contextos diferentes de interface e aplicações. Ao longo desta seção serão descritas algumas métricas julgadas importantes para medir a experiência de usuário em um contexto geral e, então, será dissertada a viabilidade de cada uma destas métricas no contexto específico do sistema de gestão de redes desenvolvido.

5.1.1 Taxa de Conversão

A taxa de conversão de um sistema representa a taxa de usuários que efetivamente se cadastram em um determinado sistema em relação ao número total de usuários que acessaram tal sistema. Esta métrica é de grande importância para sistemas comerciais que visam ter uma grande base de usuários, como é o caso de muitas redes sociais e sistemas de vendas. No caso do sistema de gestão de rede esta métrica não é aplicável, uma vez que este sistema foi desenvolvido para o uso interno em uma instituição, sendo assim, a taxa de conversão seria de 100%, o que não demonstra de maneira realista a experiência dos usuários deste sistema.

5.1.2 *Net Promoter Score (NPS)*

Net Promoter Score é um termo do inglês que pode ser traduzido¹ para pontuação de chance de promoção. Esta métrica analisa a experiência que os usuários de um determinado sistema tiveram e, a partir desta experiência, determina uma nota para cada um destes usuários. A nota de um usuário representa a probabilidade deste usuário indicar positivamente, não indicar ou indicar negativamente (ativamente falar mal sobre) o sistema ou interface em questão. O mesmo problema apresentado em 5.1.1 aparece na avaliação desta métrica para o sistema de gestão de rede, uma vez que este sistema não é um sistema comercial, não se aplicando a necessidade de um usuário indicar o sistema para demais pessoas.

5.1.3 Taxa de Sucesso de Tarefas

A taxa de sucesso de uma tarefa diz respeito à quantidade de tentativas bem sucedidas da realização de uma tarefa específica de uma interface ou sistema em relação ao número total de tentativas feitas pelos usuários. Esta métrica é relevante para o sistema de gestão de redes, uma vez que este sistema possui diversas funções de complexidade de execução elevadas, além de um fluxo complexo da execução de eventos de criação e edição de elementos por meio da interface de mapa. É de extrema importância para o desenvolvimento do sistema descobrir se as funcionalidades implementadas são intuitivas e simples para garantir a melhor experiência para o usuário.

5.1.4 Escala de Usabilidade do Sistema

A escala de usabilidade do sistema- ou *System Usability Scale*- é uma métrica utilizada para avaliar a diversos aspectos da usabilidade de um sistema. Os principais critérios avaliados pela escala de usabilidade do sistema são: a efetividade, a eficiência e a satisfação. A efetividade de um sistema avalia se os usuários conseguem realizar as tarefas desejadas; a eficiência permite avaliar o nível de esforço necessário para a realização das tarefas; a satisfação representa justamente o quão satisfeito o usuário está com o sistema em questão.

Como descrito em Lewis[15](2009), a escala de usabilidade de sistema consiste em dez itens:

1. Eu gostaria de utilizar este sistema com frequência;
2. Eu achei o sistema desnecessariamente complexo;

¹Tradução livre.

3. Eu achei o sistema fácil de usar;
4. Eu acho que precisaria do auxílio de um técnico para conseguir utilizar este sistema;
5. Eu acho que as funções do sistema foram bem integradas;
6. Eu acho que o sistema é muito inconsistente;
7. Eu imagino que a maioria das pessoas aprenderiam a utilizar este sistema rapidamente;
8. Eu achei o sistema incômodo de usar;
9. Eu me senti confiante utilizando este sistema;
10. Eu precisei aprender diversas coisas novas para conseguir utilizar este sistema.

Estas dez afirmativas devem ser avaliadas pelo usuário em notas de 1 a 5. A nota 1 representa uma forte discordância do usuário com a afirmativa e a nota 5 representa um forte concordância.

A partir da avaliação das afirmativas pelo usuário, é possível calcular a escala de usabilidade do sistema, seguindo a seguinte lógica:

- Respostas com notas ímpares (1, 3 e 5) devem ser subtraídas em 1 unidade;
- Respostas com notas pares (2 e 4) devem ser subtraídas de 5, ou seja, caso a nota do usuário seja 4, deve ser contabilizado o resultado como 1;
- Os resultados obtidos nas dez afirmativas, após terem sido efetuadas as contas mencionadas, deve ser somados e multiplicados por 2,5.

Calculada a nota a partir da lógica descrita, é possível então realizar a análise do sistema de maneira crítica. A nota média do *System Usability Scale* é 68, ou seja, caso a nota de um sistema esteja abaixo deste valor, é possível que este sistema possua grandes problemas com sua usabilidade. Em contrapartida, se um sistema obtiver uma nota muito acima do valor médio, significa que a usabilidade deste sistema é ótima.

5.1.5 Escala de Dificuldade de Tarefa

A escala de dificuldade de tarefa é uma métrica utilizada para avaliar a dificuldade do usuário para realizar determinadas tarefas. A execução desta métrica consiste em realizar questionários com os usuários de um determinado sistema, coletando notas atribuídas por estes em relação à dificuldade que estes usuários tiveram ao realizar a tarefa em questão. A escala utilizada pode variar dependendo do nível de especificidade desejado para as respostas. Para a avaliação deste sistema, será utilizada a escala com quatro pontos:

1. Muito difícil;
2. Difícil;
3. Fácil;
4. Muito fácil;

5.2 Avaliação do Sistema

Após o desenvolvimento do sistema e das diversas reuniões de *feedback* com o cliente, foi possível realizar o teste do sistema de maneira geral com o usuário final. Este teste foi a primeira vez que o usuário final teve acesso ao sistema em sua própria máquina, visto que o processo de *feedback* anteriormente feito era realizado por meio do compartilhamento de tela dos desenvolvedores, avaliando somente se as funcionalidades implementadas estavam de acordo com as necessidades do usuário, sem haver nenhum tipo de teste de usabilidade.

Para evitar que os resultados do teste de usabilidade sofresse qualquer tipo de influência por parte dos desenvolvedores, foi garantido o acesso ao sistema para o usuário sem nenhuma explicação prévia de como executar suas funcionalidades. Para cada etapa do teste, apenas foi solicitado ao usuário que realizasse uma tarefa específica, permitindo avaliar a intuitividade do sistema, os tipos de erro que podem aparecer ao longo da execução, bem como avaliar se a disposição dos elementos- como menus e barras de ferramentas- foi feita de maneira intuitiva.

5.2.1 Navegação

Taxa de Sucesso

O primeiro teste de usabilidade da interface desenvolvida consistiu em solicitar ao usuário que, a partir da tela inicial do sistema, navegasse para a página de importação de arquivos kml e kmz, para que esta funcionalidade também pudesse ser testada. A página de importação de arquivos é um subitem do item "Rede" do menu de navegação do sistema. A primeira dificuldade aparente do usuário foi encontrar qual dos itens do menu é o responsável pela importação de arquivos, o que ficou evidente pelo fato do usuário ter clicado em todos os itens do menu administrativo até perceber que o último item da navegação era responsável por mostrar os subitens relacionados à rede. Foram realizadas 11 tentativas de abrir a página de importação de arquivos, sendo a última tentativa a única bem sucedida. A partir destas tentativas do usuário é possível calcular a taxa de sucesso desta tarefa, obtendo um resultado de apenas 9%.

É importante ressaltar que ao decorrer do processo de teste foi possível perceber novamente a dificuldade apresentada pelo usuário ao tentar localizar no menu de navegação o item correto que deveria selecionar, constantemente levando o ponteiro de seu mouse à parte do menu relativa à parte administrativa do sistema.

Escala de Dificuldade

A nota atribuída pelo usuário para a realização desta tarefa foi 4, significando que o usuário achou a tarefa muito fácil de ser realizada, apesar deste resultado não ter sido observado em prática.

5.2.2 Importação de Arquivos

Taxa de Sucesso

Este teste de funcionalidade consistiu em pedir para o usuário realizar a importação de um arquivo de extensão kml. Ao achar no menu o item correspondente à importação de arquivos, o usuário conseguiu importar corretamente o arquivo de teste em sua primeira tentativa, obtendo uma taxa de sucesso de 100% para esta tarefa.

Escala de Dificuldade

A nota atribuída à esta funcionalidade foi 4, novamente demonstrando que o usuário achou esta tarefa muito fácil de ser realizada. Esta nota condiz com o que foi observado no processo de testes, uma vez que sua taxa de sucesso foi de 100%.

5.2.3 Interface de Tabelas

Taxa de Sucesso

A próxima funcionalidade testada foi a adição de elementos por meio da interface de tabela, presente na página dedicada às caixas de emenda. O usuário, de maneira quase instantânea, identificou onde na tabela deveria ser clicado para adicionar um novo elemento, abrindo assim o modal de criação. O preenchimento da maioria dos dados apresentados no formulário foi feita sem nenhum tipo de problema, com a exceção dos campos de latitude e longitude, nos quais o usuário questionou se não apareceria um mapa para selecionar o ponto desejado para a criação do elemento. Ao preencher manualmente os campos de latitude e longitude, o usuário preencheu os dados- tanto latitude quanto longitude- como sendo coordenadas positivas, o que fez a caixa de emenda em questão ser criada em uma localização distante da localização real desejada. O mesmo teste foi repetido para outros elementos, e a resposta do usuário foi igual em todas as situações apresentadas. Apesar

desta funcionalidade ter sido executada sem nenhum tipo de erro do sistema, o preenchimento errôneo das coordenadas geográficas (por problemas de interface) fez com que a taxa de sucesso desta tarefa caísse para 33%, visto que o usuário preencheu de maneira errada dois dos seis campos presentes no formulário.

Após a criação de elementos por meio das tabelas, o usuário foi solicitado testar a exclusão e edição, ainda na mesma interface de tabelas. Ambos testes foram concluídos sem nenhum erro do sistema ou do usuário, o que também acarretou em uma taxa de sucesso de 100%. Outra funcionalidade que o usuário conseguiu utilizar sem nenhuma explicação prévia- e com taxa de sucesso 100%- foi a pesquisa por campos na interface de tabela, mencionada na Seção 4.2.

Escala de Dificuldade

Para as tarefas de manipulação de dados por meio da interface de tabela, o usuário atribuiu uma nota 4. Esta nota condiz com o que foi observado nos testes, visto que o usuário apresentou leves dificuldades no preenchimento de informações.

5.2.4 Interface de Mapa

Taxa de Sucesso

Ao acessar a página de mapa, com caixas de emenda e trechos de cabo já existentes devido a importação do arquivo kml, o usuário rapidamente descobriu a funcionalidade de mostrar informações adicionais de um elemento ao clicar no mesmo. Apesar de ter encontrado esta funcionalidade ao clicar em caixas de emenda, o usuário teve dificuldades em realizar o mesmo processo para trechos de cabo, uma vez que a linha compõe este elemento possui uma espessura pequena, dificultando o clique neste tipo de elemento. Por este motivo, o usuário presumiu que este tipo de elemento não era capaz de apresentar informações adicionais. O cálculo da taxa de sucesso para esta tarefa é um pouco complexo, uma vez que a funcionalidade foi descoberta pelo usuário em sua primeira tentativa, porém não foi facilmente reproduzida ao tentar fazer o mesmo em outro tipo de elemento. Tendo esta peculiaridade em mente, a taxa de sucesso final desta funcionalidade foi calculada em 40%.

O primeiro teste guiado para a interface de mapa consistiu em solicitar ao usuário que realizasse a adição de um elemento do tipo marcador (caixa de emenda ou poste) por meio da interface de mapa. O usuário demonstrou grande dificuldade em selecionar qual tipo de elemento deseja adicionar ao mapa por meio da barra de ferramentas (presente na Figura 4.5). Devido esta dificuldade, a funcionalidade de seleção de tipo de elemento geográfico a ser criado obteve uma taxa de sucesso de aproximadamente 38%, visto que

o usuário realizou 8 tentativas de selecionar o tipo de elemento e apenas 3 vezes o mesmo conseguiu selecionar o tipo desejado.

Sabendo como iniciar a criação de um elemento no mapa, o usuário então começou o teste de criação selecionando o tipo caixa de emenda. Este teste, assim como o teste de criação de postes por meio da interface de mapa foi feito com sucesso por parte do usuário e do sistema, tendo uma taxa de sucesso de 100%.

O mesmo teste feito para caixas de emenda e poste foi repetido para criação de trechos de cabo, que não utiliza o mesmo botão de criação dos elementos de ponto geográfico. O usuário naturalmente achou o botão responsável pela seleção de criação deste tipo de elemento e intuitivamente concluiu a criação de um trecho de cabo sem nenhum erro, alcançando uma taxa de sucesso de 100%.

Para os teste subsequentes, foi solicitado ao usuário que editasse e excluísse elementos visíveis na página de mapa. O usuário rapidamente localizou na barra de ferramentas a opção responsável pela edição de elementos e conseguiu em sua primeira tentativa realizar uma edição. O mesmo sucesso obtido no teste de edição foi alcançado no processo de deleção de elementos. Ambos testes alcançaram taxa de sucesso de 100%.

Escala de Dificuldade

Nesta parte da avaliação, o usuário atribuiu nota 5 em relação à facilidade de criação de elementos pela interface de mapa, mas atribuiu nota 4 para as funcionalidades de edição e exclusão de elementos. Este resultado não foi esperado, visto que o usuário apresentou mais dificuldade no processo de criação do que no processo de edição e exclusão.

5.2.5 Avaliação da Escala de Usabilidade

Ao finalizar a execução de todos os testes solicitados, o usuário foi capaz de avaliar o sistema de gestão de redes, respondendo com notas entre 1 e 5 as dez afirmativas do *System Usability Scale*, enumeradas na Seção 5.1.4. Na Tabela 5.1 são apresentados os valores atribuídos pelo usuário na coluna da esquerda, seguidos pelos valores calculados baseados na lógica explicada na Seção 5.1.4:

Afirmativa	Nota do Usuário	Nota Calculada
1	5	4
2	2	3
3	5	4
4	3	2
5	4	1
6	1	0
7	5	4
8	2	3
9	4	1
10	1	0

Tabela 5.1: Notas referentes às afirmativas de escala de usabilidade do sistema. A coluna esquerda apresenta o número da afirmativa em questão. A coluna do meio apresenta os valores atribuídos pelo usuário. A coluna direita apresenta os valores obtidos após a aplicação da lógica mencionada na Seção 5.1.4.

A partir das notas atribuídas pelo usuário, o cálculo de pontuação da escala de usabilidade do sistema foi feito, obtendo a pontuação 55. Como foi explicado na Seção 5.1.4, uma nota abaixo de 68 geralmente significa que o sistema avaliado possui problemas com sua usabilidade. A partir da avaliação da taxa de sucesso de tarefas calculada ao longo da Seção 5.2 é possível entender quais funcionalidades podem ter levado a nota de escala de usabilidade do sistema para um número relativamente baixo. Funcionalidades como a navegação e a seleção de tipo de elemento na interface de mapa devem ter grande influência na avaliação final do usuário, o que reforça a ideia de que estas funcionalidades e decisões de *design* tomadas para a implementação destas devem ser retrabalhadas para garantir uma melhor experiência ao usuário.

5.2.6 Erros Encontrados

Apesar de possuírem uma alta taxa de sucesso, a funcionalidade de adição de elementos do tipo marcador evidenciaram uma falha de design de interface, devido a baixa taxa de sucesso da funcionalidade de seleção de tipo de elemento. O usuário encontrou problemas ao tentar criar dois postes em sequência, uma vez que o sistema sempre retorna à seleção de tipo de elemento para caixa de emenda ao finalizar a criação de um elemento do tipo marcador. Esse *feedback* evidenciou a necessidade de não selecionar nenhum tipo de elemento por padrão, deixando a escolha sempre ser do usuário, além de manter a última escolha feita, permitindo um melhor fluxo de criação de múltiplos elementos.

5.3 Resumo Conclusivo

Os resultados obtidos nesta seção foram úteis na identificação de problemas de escolhas no âmbito do design de interface, assim como na verificação de funcionamento do sistema em si. As funcionalidades implementadas não geraram nenhum tipo de erro de funcionamento, garantindo uma usabilidade sem problemas para o usuário. Apesar disto, as decisões tomadas em relação ao *design* do menu de navegação e da barra de ferramenta referente à seleção de tipo de elemento apresentaram uma baixa taxa de sucesso, evidenciando a necessidade de alterar a maneira como estas funcionalidades são dispostas ao usuário final.

A taxa de sucesso de uma tarefa mostrou-se uma métrica pertinente e eficaz para a quantificação da experiência de usuário, considerando o contexto do sistema no que diz respeito à quantidade de usuários e à natureza não comercial do sistema desenvolvido. O processo de teste dos diferentes tipos de interface provou ser útil no ato de levantamento de melhorias para o sistema. Os *feedbacks* do usuário, bem como os erros de design percebidos pela dificuldade de execução de determinadas tarefas por este ao longo do teste, geraram dados suficientes para que seja considerado relevante uma refatoração de partes do código do sistema, bem como melhorias no fluxo de criação e edição de elementos e a alteração de ícones de difícil reconhecimento pelo usuário.

A escala de usabilidade do sistema foi capaz de representar a dificuldade do usuário com as funcionalidades do sistema apresentadas a ele. A nota calculada para a usabilidade por meio desta métrica reforça a necessidade de refatoração e alteração do *design* da interface.

Por fim, a escala de dificuldade de tarefa também se provou ser uma métrica efetiva na quantificação da experiência do usuário. Ocorreram algumas inconsistências nas respostas dadas pelo usuário em relação ao que foi observado no teste de funcionalidades feito por este, mas ainda foi possível observar uma relação entre esta métrica e as demais, reforçando as falhas de interface já mencionadas.

Capítulo 6

Conclusão

A implementação de design centrado no usuário para o desenvolvimento de interfaces é uma excelente escolha para tentar garantir uma melhor experiência aos usuários de um sistema. A utilização de técnicas como *feedback* constante por meio do usuário final, bem como sessões de teste em um ambiente controlado, são excelentes práticas presentes no cenário de desenvolvimento *front-end*. Além de uma melhor usabilidade, o design centrado no usuário- quando bem implementado- também pode reduzir a quantidade de correções e ajustes feitos ao longo do desenvolvimento, reduzindo o tempo gasto para a realização de um projeto, reduzindo custos com o sistema, permitindo uma melhor alocação de tempo da equipe de desenvolvimento, podendo esta focar na implementação de novas funcionalidades.

Com o constante crescimento do uso de *smartphones* e *tablets* para a realização dos mais diversificados tipos de tarefas, garantir uma melhor experiência de usuário é cada vez mais fundamental. Possuir o conhecimento necessário para desenvolver uma interface prática e útil pode ser o diferencial de um sistema em um meio competitivo, como é o caso para sistemas de compras *online*, ou até mesmo para manter um alto engajamento do usuário por uma maior quantidade de tempo em uma determinada aplicação, como é o caso das redes sociais.

O sistema de gestão de redes desenvolvido para suprir a necessidade da instituição GigaCandanga, alcançou o objetivo esperado: facilitar o processo de gestão das redes de fibra ótica e todos os elementos que a compõem. A interface desenvolvida se apresentou capaz de realizar os requisitos definidos pelo usuário e pelo cliente ao início do projeto. O sistema apresenta diversos elementos por meio de uma interface interativa de mapa, além de possuir páginas de criação, edição e deleção para todos os elementos que fazem parte da rede.

A implementação destes requisitos foi avaliada pelo usuário por meio de uma sequência de testes de usabilidade descritos na Seção 5.2. Este teste foi analisado e, a partir disto,

foram utilizadas métricas de experiência de usuário para ser possível quantificar a usabilidade e a própria experiência que o usuário teve durante o uso do sistema. As métricas utilizadas para a quantificação da experiência de usuário se provaram muito pertinentes, além de seus resultados serem condizentes entre si.

A partir do estudo descrito ao longo deste documento, foi possível perceber quais práticas de design de interface e quais decisões de projeto foram eficazes na criação de uma interface capaz de proporcionar uma melhor experiência de usuário. Além de boas práticas, foi possível, a partir da observação da utilização do sistema por seu usuário, determinar quais decisões de design tomadas não foram intuitivas, bem como fluxos de cadastro que deveriam ter sido feitos de maneira mais eficiente e simplificada. A partir dos resultados obtidos, tanto em relação à experiência de usuário quanto em relação à quantidade de erros de sistema obtidos, ficou evidente que as escolhas feitas em relação às tecnologias utilizadas fizeram sentido no contexto da aplicação em questão, sendo capaz de desenvolver em tempo reduzido uma interface suficientemente robusta para lidar com a grande quantidade de elementos visuais necessários.

Referências

- [1] *Gigacandanga*. <https://gigacandanga.net.br/>. Acesso em: 22/10/2020. 1
- [2] CHAVES, João. Sistema de monitoramento de redes de fibra óptica: Desenvolvimento e projeto voltados à experiência do usuário. Orientador: André Drummond. 2020. 49 f. Monografia (Graduação) – Engenharia de Computação, CIC, UnB, Brasília. 2020. 1
- [3] DAMACENO, Andre. O GERENCIAMENTO CADASTRAL DE UMA REDE DE FIBRAS ÓPTICAS UTILIZANDO UMA API REST: MODELAGEM E ARQUITETURA. Orientador: André Drummond. 2020. 48 f. Monografia (Graduação) – Engenharia de Computação, CIC, UnB, Brasília. 2020. 1
- [4] JORGE, Lucas. PROJETO E ARQUITETURA DE API REST PARA SISTEMA DE MONITORAMENTO DE REDES ÓPTICAS. Orientador: André Drummond. 2020. 43 f. Monografia (Graduação) – Engenharia de Computação, CIC, UnB, Brasília. 2020. 1
- [5] Gorelick, Noel, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau e Rebecca Moore: *Google earth engine: Planetary-scale geospatial analysis for everyone*. Remote Sensing of Environment, 2017. <https://doi.org/10.1016/j.rse.2017.06.031>. 3
- [6] Lowdermilk, Travis: *User-centered design: a developer's guide to building user-friendly applications*. " O'Reilly Media, Inc.", 2013. 7
- [7] Albert, William e Thomas Tullis: *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013. 8
- [8] Walke, Jordan: *React - uma biblioteca javascript para criar interfaces de usuário*, 2013. <https://pt-br.reactjs.org/>. 9
- [9] Agafonkin, Vladimir: *Leaflet- a javascript library for interactive maps*, 2010. <https://leafletjs.com/>. 11
- [10] Beck, Kent, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries *et al.*: *Manifesto for agile software development*. 2001. 12
- [11] Schwaber, Ken e Mike Beedle: *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River, 2002. 12

- [12] Microsoft: *Microsoft excel*, 2016. <https://www.microsoft.com/pt-br/microsoft-365/excel>. 15
- [13] Wallace, Evan e Dylan Field: *Figma, the collaborative interface design tool*, 2015. <https://www.figma.com/>. 17
- [14] *Ícone de bússola do material design*. <https://material.io/resources/icons/?search=compas&icon=explore&style=baseline>. 25
- [15] Lewis, James R e Jeff Sauro: *The factor structure of the system usability scale*. Em *International conference on human centered design*, páginas 94–103. Springer, 2009. 33